

from ng-valid to ng-invalid.

Day-16
16/04/2020

class-name property:- class name property stores all properties

form-control, ng-touched, ng-dirty, ng-pristine.

```
<div class="form-group">
```

```
<label> Name </label>
```

```
<input type="text" #name="ngModel" class="form-control"
      name="username" [(ngModel)]="userModel.username"
      required>
```

```
</div>
```

```
{{ name.className }}
```

```
untouched : {{ name.untouched }} <br>
```

```
pristine : {{ name.pristine }} <br>
```

```
invalid : {{ name.invalid }} <br>
```

Output:-

Before:- Name

untouched : true

pristine : true

invalid : false

After:- Name

untouched : false

pristine : false

invalid : true

```
<input type="text" class="form-control is-invalid"
      name="username" #name="ngModel"
      [(ngModel)]="userModel.username" required>
```

Output:-

Name

Bushma

```
<input type="text" class="form-control" name="username"
      #name="ngModel" [class.is-invalid]="name.invalid"
      [(ngModel)]="userModel.username" required>
```

invalid: {{ name.invalid }}

O/p:-

Before

Name

Bushma

invalid: false

After:-

Name

invalid: true

```
<input type="text" class="form-control" name="username"
      #name="ngModel" [class.is-invalid]="name.invalid"
      & #name.touched" [(ngModel)]="userModel.username"
      required>
```

invalid: {{ name.invalid }}

untouched: {{ name.untouched }}

O/p:-

Before

Name

invalid: true

untouched: true

After

Name

invalid: true

untouched: false

```

<div class="form-group">
  <table> Phone Number </table>
  <input type="text" class="form-control" name="phone"
    #phone="ngModel" [class.is-invalid]="phone.invalid &&
    phone.touched" [(ngModel)]="userModule.phone" required>
</div>

```

Output:-

Phone Number

Phone Number

Just touched & did not
write the content

Pattern:-

```

<input type="text" class="form-control" name="phone"
  #phone="ngModel" pattern="\d{10}$" [class.is-
  invalid]="phone.invalid && phone.touched"
  [(ngModel)]="userModule.phone" required>

```

Output:-

Phone Number

 → Invalid phone number

```

<input type="text" class="form-control" name="phone"
  #phone="ngModel" pattern="^[7-9]{1}[0-9]{9}$"
  [class.is-invalid]="phone.touched" && phone.
  invalid" [(ngModel)]="username.phone" required>

```

Output:-

Phone Number

 → valid number starting
with 7/8/9 having 10-digits


```
<div class="form-group">
```

```
<label>Email</label>
```

```
<input type="text" class="form-control" name="email"
```

```
pattern="[a-z]{4,15}(@gmail.com)" #email="ngModel"
```

```
[class="{{ !isValid }}" email.invalid && email.touched"
```

```
[(ngModel)]="userModel.email" required>
```

O/p:-

Email

abcd12@yahoo.

Email

abcd123@gmail.com

* ngModel is a directive which is used for text, selection etc.

* It stores the required user value to a template reference variable.

* we can use that variable (name, email, phone no)

* It is also used for validation in a form.